CENG 383 Real-Time Systems 2023-2024 Spring
Term Project

Implementation of task (thread) mechanism, time management functions and a scheduler.

Implement the following mechanisms and functions in OSx383 real-time nano kernel.

1. Task Mechanism:

Implement necessary functions to create tasks/threads. The following code will help you.

```
void Task1(void *arg)
{
  set_timer_periodic( … )

  while(1)
 {
    // do something
    wait_timer_event()
 }
}

void Task2(void *arg))
{
 set_timer_periodic( … )

  while(1)
 {
    // do something
    wait_timer_event()
 }
}

void Task3(void *arg))
{
  while(1)
 {
    // do something

 }
}

// you can define more tasks

main() {
...

  tasks_t  t1, t2;
  t1.stack_size=256
  t1.period=40000   // in microseconds
  t1.priority=4

  task_create(&t1, &Task1)

...

}
------------------------------
```

If the period of a task is non-zero than the task is treated as a periodic task. Otherwise the task is an aperiodic task.

You can implement your code in main.c or other relevant source files.

set_timer_periodic(.???.) function sets up the timer so that periodic timer events occur (period is an argument that should be passed to this function).
wait_timer_event() suspends the task and calls the scheduler. The task is put back to the ready queue as soon as the timer event occurs.

2. Time Management

Besides set_timer_periodic() function, you need to implement a basic time management mechanism such that your timing algorithm fed by timer interrupt, creates timer events for the tasks. The timing algorithm should know all the task parameters, particularly the period parameter of each task, and creates the events accordingly.

3. The scheduler

Scheduler will be preemptive and priority based scheduler. The scheduler should run in a round-robin fashion. The fixed quantum time is 10 ms.

For any questions, don’t hesitate to ask me!